

数独の複数候補のリスト candlist  
文字列の修正 subsn - J602 版に向けて  
Jコード探訪 - 2

西川 利男

数独をやっているとき、1つのセルに入る候補の数字が複数ある場合がよくある。セルの位置を指定して、その候補を表示するには関数 cand を使って示すことができる。

例として、次の毎日新聞の問題で行う。

```
A0 =: main_1026
```

```
see A0
```

```
+---+---+---+
|.5. |...|.94|
|..1|..2|..8|
|...|8..|5..|
+---+---+---+
|..5|.2. |.3. |
|8..|9.4|..2|
|.6. |.7. |1.. |
+---+---+---+
|..2|..9|...|
|1..|5..|7..|
|67. |...|.4. |
+---+---+---+
```

セルの位置に対して、たとえば次のようにする。ここでは1オリジンで示すことにする。

```
A0 cand 1 1
```

```
2 3 7
```

```
A0 cand 1 3
```

```
3 6 7 8
```

ところで、多くのセルを、できればすべてのセルに入る数字の可能性を見ることができれば非常に便利である。

これを表示するツールをプログラミングしてみた。そしてこのために、いくつかのコーディング・テクニックを必要とした。Jユーザにとって、興味深い問題と考え、これについて紹介する。

また、候補の頻度を見るには、関数 free を使って、次のようにすれば良い。

```
see +/"1 free A0
```

```
+---+---+---+
|3.4|434|3..|
|43.|45.|22.|
|545|.54|.44|
+---+---+---+
|33.|2.3|4.3|
|.22|.4.|13.|
|4.3|1.3|.22|
+---+---+---+
|33.|55.|344|
|.44|.43|.33|
|..3|333|4.4|
+---+---+---+
```

つまり、この表の値はセル(1, 1)では3通り、セル(1, 3)では4通りを示す。

さて、問題のプログラムを作ってみよう。原理的には簡単だが、Jの画面での表示の方法がちょっとやっかいである。

まず、すべてのインデックスを示す数値の組をつくる。

```
NN =: >: (9,9)#:>i.81
```

```
NA =: " : <"(1) NN
```

```
MM =: (9,9)$<"(1) NN
```

```
MM
```

```
+---+---+---+---+---+---+---+---+---+
|1 1|1 2|1 3|1 4|1 5|1 6|1 7|1 8|1 9|
+---+---+---+---+---+---+---+---+---+
|2 1|2 2|2 3|2 4|2 5|2 6|2 7|2 8|2 9|
+---+---+---+---+---+---+---+---+---+
|3 1|3 2|3 3|3 4|3 5|3 6|3 7|3 8|3 9|
+---+---+---+---+---+---+---+---+---+
|4 1|4 2|4 3|4 4|4 5|4 6|4 7|4 8|4 9|
+---+---+---+---+---+---+---+---+---+
|5 1|5 2|5 3|5 4|5 5|5 6|5 7|5 8|5 9|
+---+---+---+---+---+---+---+---+---+
|6 1|6 2|6 3|6 4|6 5|6 6|6 7|6 8|6 9|
+---+---+---+---+---+---+---+---+---+
|7 1|7 2|7 3|7 4|7 5|7 6|7 7|7 8|7 9|
+---+---+---+---+---+---+---+---+---+
|8 1|8 2|8 3|8 4|8 5|8 6|8 7|8 8|8 9|
+---+---+---+---+---+---+---+---+---+
|9 1|9 2|9 3|9 4|9 5|9 6|9 7|9 8|9 9|
+---+---+---+---+---+---+---+---+---+
```

一般にボックスの中の値に、それぞれ操作をするには動詞に接続詞L:0を添えて作用させるのが、常套である。つまり、このボックスに入れられたインデックスのおのにおのに、先の関数 cand をL:0で作用させる。

ところがこの結果はJの実行画面では見られるが、プリンターではプリント幅の制約で、きれいに見ることは出来ない。これについては別のレポートで述べているが、この際、必要にせまられて開発したが、ツール pr\_adj を用いて、次のようにする。

49 pr\_adj A0 cand L:0 MM

2 3 7		3 6 7 8	1 3 6 7	1 3 6	
3 4 7 9	3 4 9		3 4 6 7	3 4 5 6 9	
2 3 4 7 9	2 3 4 9	3 4 6 7 9		1 3 4 6 9	
4 7 9	1 4 9		1 6		
	1 3	3 7		1 3 5 6	
2 3 4 9		3 4 9	3		
3 4 5	3 4 8		1 3 4 6 7	1 3 4 6 8	
	3 4 8 9	3 4 8 9		3 4 6 8	
		3 8 9	1 2 3	1 3 8	

1 3 6 7	2 3 6			
	3 6	6 7		
1 3 6 7		1 2 6 7	1 3 6 7	
1 6 8	4 6 8 9		6 7 9	
	6	5 6 7		
3 5 8		5 8	5 9	
	3 6 8	1 5 6 8	1 3 5 6	
3 6 8		2 6 8	3 6 9	
1 3 8	2 3 8 9		1 3 5 9	

-----+-----+-----+-----+

このようにして、目的は達せられたが、もうすこし簡便にしたいものである。  
 そのために今ひとつの工夫をしたのが、以下の関数 candlist である。

結果は次のようになる。

```

candlist A0
+-----+
|+---+---+---+|+---+---+---+|+---+---+---+| | | | | | | | |
|2 3|  |3 6||1 3|1 3|1 3||2 3|  |  |
|7 |  |7 8||6 7|6 |6 7||6 |  |  |
+-----+
|3 4|3 4|  ||3 4|3 4|  ||3 6|6 7|  |
|7 9|9 |  ||6 7|5 6|  ||  |  |  |
+-----+
|  |9 |  |  |+-----+
|2 3|2 3|3 4|+---+---+---+|  |1 2|1 3|
|4 7|4 9|6 7||  |1 3|1 3||  |6 7|6 7|
|9 |  |9 |  ||  |4 6|6 7|+-----+
+-----+
|  |9 |  |  |
|  |  |+-----+
+-----+
+-----+
|+---+---+---+|+---+---+---+|+---+---+---+| | | | | | | | | |
|4 7|1 4|  ||1 6|  |1 6||4 6|  |6 7|
|9 |9 |  ||  |  |8 |  ||8 9|  |9 |
+-----+
|  |1 3|3 7||  |1 3|  ||6 |5 6|  |
|  |  |  ||  |5 6|  ||  |7 |  |
+-----+
|2 3|  |3 4||3 |  |3 5||  |5 8|5 9|
|4 9|  |9 |  ||  |  |8 |  ||  |  |  |
+-----+
+-----+
+-----+
|+---+---+---+|+---+---+---+|+---+---+---+| | | | | | | | |
|3 4|3 4|  ||1 3|1 3|  ||3 6|1 5|1 3|
|5 |8 |  ||4 6|4 6|  ||8 |6 8|5 6|
+-----+
|7 |8 |  |  |+-----+
|  |3 4|3 4|+---+---+---+|  |2 6|3 6|
|  |8 9|8 9||  |3 4|3 6||  |8 |9 |
+-----+
|  |6 8|8 |  |+-----+
|  |  |3 8|+---+---+---+|2 3|  |1 3|
|  |  |9 |  ||1 2|1 3|1 3||8 9|  |5 9|
+-----+
|3 |8 |8 |  |+-----+
|  |  |+-----+
+-----+

```

NB. candidate list 2010/11/27 =====

```
NN =: >: (9,9)#:>i.81
NA =: " : <"(1) NN
MM =: (9,9)$<"(1) NN
subs=: [. & (((e.&) ((# i.@#)@)) (@))] }
candx =: 3 : 0
:
CC =. x. cand y.
(' ' subs ' _')"(0) " : ((2 >. >. -: # CC), 2)$CC, 3#_
)
NB. e.g. candlist A0
candlist =: 3 : 0
(3 3, : 3 3)&<(. 3) y. candx L:0 (9,9)$<"(1) >: (9,9)#:>i.81
)
```

つまり、ボックスの中の値をベクトルではなく、2列の配列として表示するようにした。また、小さなことであるが、値の要素数が1や3の場合にも2列とするため、空白の代わりに無限大(∞)を追加して配列を作った。その後、文字化してアンダー・文字(∞)を空白に置き換えた。ここで、置き換えには、Jのイディオム subs を用いた。J305, J402ではこれで、めでたしであった。

この際、数独プログラム nhui\_sud.ijs を J602 で動く版を作ることにした。動詞の引数の x. や y. さらに副詞や接続詞の m. や n. などの置き換えは何なく行った。ところが、J602での難物が起きた。J602では subs で使っている [. がサポートされていない！

```
subs=: [. & (((e.&) ((# i.@#)@)) (@))] }
この tacit 定義を理解しようとしたが容易ではない。しかし、操作の意味を考えてかつ explicit 定義で行えばそれほどのことはない。次のような代替の subsn を定義して解決できた。
subsn =: 1 : 0 NB. defined by T. Nishikawa
:
x ((y e. m) # (i.@#y)) } y
)
```

これを使って、candx は以下のように定義し直した。

```
candx =: 3 : 0
:
CC =: x cand y
CCC =: " : ((2 >. >. -: # CC), 2)$CC, 3#_
' ' (' _ subsn)"(0) CCC NB. revised using subsn for J602
)
```